AUTOMATIC DISCOVERY OF NETWORK NODE ADDRESSES

CROSS–REFERENCE TO RELATED PATENT APPLICATIONS

[001] This present application claims priority to several co-pending U.S. provisional applications that were all filed on June 24, 2002 of which is incorporated by reference in its entirety herein. The co-pending U.S. provisional applications are:

[002] 60/391,098 – "Auto Topology Discover Method for Layer 3 Networks"

[003] 60/391,121 – "Method for Automatic Discovery of Network Core Type"

[004] 60/391,053 – "Method for Determination of Virtual Circuit Characteristics in Layer 3 Networks"

FIELD OF THE INVENTION

[005] The present invention relates generally to computer network architecture, and more particularly, to discovery of addresses for nodes on a computer network.

DESCRIPTION OF THE RELATED ART

[006] In order for two devices to communicate with each other over a computer network, two conditions must be met. One, a communication path must be provided which links the two nodes, a condition known as "reachability." Two, the first device must know the network address of the second device, and vice-versa. The first condition is easily met as a result of the design of the network itself and the configuration of the network by the network administrator. The second condition can be met by a variety of mechanisms.

[007] One way is for the network administrator to provide each device with its own list of network addresses of reachable nodes. This could be done through a configuration file, a command line interface, or a network management system such as SNMP. However, this mechanism is not feasible for networks which contain dozens or even hundreds of nodes.

[008] Another way is for a device to advertise its own network address to other reachable nodes by transmitting on the network using a broadcast or multicast address. A

broadcast address is a single well-known address which all network nodes are aware of and can listen on. A multicast address is one of a group of well-known addresses which all network nodes are aware of and can listen on. Therefore, a node listening on either the broadcast address or one of the multicast addresses can learn the address of the sender without knowing the sender's address ahead of time. However, for a large scale network with hundreds of nodes broadcasting, this mechanism places a heavy burden on the nodes which receive large numbers of broadcast packets.

[009] In a variation on the first mechanism, the network administrator maintains only one list of network addresses for reachable nodes, on a particular "directory services" device. Other devices then contact the directory services device to discover the network addresses of other nodes.

[010] Each of these solutions is appropriate for a different environment. The first solution is very simple to implement and works for a very small number of nodes. The second solution, implemented by protocols such as RIP (Routing Information Protocol) and OSPF (Open Shortest Path First), is used as a discovery mechanism between routers. The last solution, implemented by protocols such as LDAP (Lightweight Directory Access Protocol) and DNS (Domain Name Service), is used as a discovery mechanism for applications, such as Windows Explorer, to discover information about users, servers, printers, and other network devices.

[011] In yet another environment, the network has a large enough number of nodes to make the first mechanism (each device maintaining its own list of reachable nodes) not feasible. In this environment, an increase in broadcast traffic is deemed to be undesirable. Finally, this environment supports the execution of a software application on various nodes, and within this software application, it is undesirable to provide "full mesh" connectivity such that each node communicates with each other node. Within this software, it is instead desirable to allow certain nodes to communicate with certain other nodes, in a manner which is controllable by the network administrator. A directory services solution is more complicated than necessary in this particular environment.

[012] Therefore, there is a need for improved systems and methods which address these and other shortcomings of the prior art.

SUMMARY OF THE INVENTION

[013] The present invention provides systems and methods for automatic discovery of network addreses. Briefly described, in architecture, one embodiment of the apparatus, among others, comprises: announcer logic; listener logic; and forwarder logic. The announcer logic is configured to transmit a node address and a forward counter associated with each known node in a list, if the forward counter is greater than zero, to all nodes in the list having a static type. The listener logic is configured to receive an announcement packet and to add to the list of known nodes at least one new node. The node address and the forward counter of the new node correspond to the announcement packet, and the new node has a discovered type. The forwarder logic is configured to transmit the node address and the forward counter associated with the new node, if the forward counter is greater than zero, to all known nodes in the list.

[014] One embodiment of a method, among others, can be broadly summarized by the following steps: initializing a known node list; transmitting to all known nodes, a node address and a forward counter associated with each known node, if the forward counter is greater than zero; receiving from the network an announcement packet; adding to a list of discovered nodes at least one new discovered node, where the discovered node comprises a node address and a forward counter corresponding to the announcement packet; and transmitting to all known nodes and all discovered nodes, the node address and the forward counter associated with each known node, if the forward counter is greater than zero.

[015] Other systems, methods, features, and/or advantages of the present invention will be or become apparent to one with skill in the art upon examination of the following drawings and detailed description. It is intended that all such additional systems, methods, features, and/or advantages be included within this description, be within the scope of the present invention, and be protected by the accompanying claims.

DESCRIPTION OF THE DRAWINGS

[016] FIG. 1 is a block diagram of a communications environment where an embodiment of the system and/or method of the present invention operates.

[017] FIG. 2 is a block diagram of one embodiment of a node from FIG. 1 which utilizes an embodiment of a system and/or method for automatic discovery of network addresses.

[018] FIGs. 3A-C are a sequence of diagrams showing how the list of known nodes in each node is updated by announcements and forwarding in one example network configuration.

[019] FIG. 4 is a flowchart of an embodiment of method for automatic discovery of network addresses.

[020] FIG. 5 is a flowchart of another embodiment of method for automatic discovery of network addresses.

[021] FIGs. 6A-D are a sequence of diagrams showing how the list of known nodes in each node is updated by announcements and forwarding in another example network configuration.

[022] FIGs. 7A-D are a sequence of diagrams showing how the list of known nodes in each node is updated by announcements and forwarding in yet another example network configuration.

[023] FIG. 8 shows the structure of an announcement packet used in one embodiment of a system and method for automatic discovery of network nodes.

[024] FIGs. 9A-B are a sequence of diagrams showing how the list of known nodes in each node is updated by announcements and forwarding in another embodiment of a system for automatic discovery of network addresses.

DETAILED DESCRIPTION

[025] Systems and methods for automatic discovery of network node addresses are provided. As will be described in more below, some embodiments of the systems and methods rely on the initial provisioning of a relatively small number of node addresses by the network administrator, combined with ability of the nodes to propagate this initial set of node addresses throughout the network using unicast addressing. The discovery by one node of another node's address, either automatically or through provisioning, results in the creation of a "path" between the two nodes. By varying the initial provisioning of node addresses, the network administrator can create a network configuration comprising of paths between all nodes (a "full mesh"), between tiers of nodes, or any other

combination. Thus, applications executing on these nodes communicate with other nodes only through paths which are controllable by the network administrator.

[026] The patent application with attorney docket number 61607–1730, entitled "Automatic Discovery of Network Core Type", and filed the same day is incorporated by reference in its entirety herein. In addition, the patent application with attorney docket number 61607–1740, entitled "Determination of Network Performance Characteristics," and filed the same day is incorporated by reference in its entirety herein.

[027] The systems and/or methods can be implemented in software, hardware, or a combination thereof. In some embodiments, the system and/or method is implemented in software that is stored in a memory and that is executed by a suitable microprocessor (uP) situated in a communications device. However, system and/or method, which comprises an ordered listing of executable instructions for implementing logical functions, can be embodied in any computer-readable medium for use by or in connection with an instruction execution system, apparatus, or device, such as a computer-based system, processor-containing system, or other system that can fetch the instructions from the instruction execution system, apparatus, or device and execute the instructions.

[028] In the context of this document, a "computer-readable medium" can be any means that can contain, store, communicate, propagate, or transport the program for use by or in connection with the instruction execution system, apparatus, or device. The computer readable medium can be, for example but not limited to, an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system, apparatus, device, or propagation medium. More specific examples (a nonexhaustive list) of the computer-readable medium would include the following: an electrical connection (electronic) having one or more wires, a portable computer diskette (magnetic), a random access memory (RAM) (magnetic), a read-only memory (ROM) (magnetic), an erasable programmable read-only memory (EPROM or Flash memory) (magnetic), an optical fiber (optical), and a portable compact disc read-only memory (CDROM) (optical). Note that the computer-readable medium could even be paper or another suitable medium upon which the program is printed, as the program can be electronically captured, via for instance optical scanning of the paper or other medium, then compiled, interpreted or

otherwise processed in a suitable manner if necessary, and then stored in a computer memory.

[029] FIG. 1 is a block diagram of a communications environment in which an embodiment of the system and/or method for automatic discovery of network nodes operates. The computer network 100 comprises one or more computer systems, known as network nodes 101a-d, in communication with each other through routed network 102. Each node 101a-d has a network address 103a-d, and a link 104a-d to routed network 102. The systems and methods for automatic discovery of network nodes are applicable for nodes 101a-d of many different types, including but not limited to personal computers and workstations, computer peripherals such as printers and scanners, and network devices such as hubs, bridges, routers, and switches. Routed network 102 may be any network which provides layer 3 (network protocol layer) connectivity. Although the discussion herein makes reference to Internet Protocol (IP) as the network protocol, the systems and methods for automatic discovery of network nodes are not limited to IP, but is applicable to any network protocol that supports unicast network addresses. Many different types of links 104 can be used to connect each node 101 to routed network 102, including but not limited to analog telephone lines, DSL, Ethernet, and wireless. The systems and methods for automatic discovery of network nodes are not dependent on any particular type of link 104.

[030] A node 101a is "reachable" by another node 101b if the combination of links 104a,b and routed network 102 provides a communication path between the two nodes. However, one node 101a can communicate with another reachable node 101b only if node 101a knows the network address 103b of the reachable node 101b. If node 101a does know the network address 103b of the reachable node 101b, then a "path" exists between node 101a and node 101b.

[031] Paths may be unidirectional or bidirectional. If node 101a knows the network address 103b of node 101b but node 101b does not know the network address 103a of node 101a, then there is one unidirectional path <u>from</u> node 101a <u>to</u> node 101b. However, if node 101a knows the network address 103b of node 101b, and node 101b also knows the network address 103b of node 101b, then there is also a second unidirectional path,

from node 101b to node 101a. These two unidirectional paths, taken together, may also be referred to as a single birectional path between node 101a and node 101b.

[032] In FIG. 1, one unidirectional path 105 and one bidirectional path 106 are shown. Unidirectional path 105 implies that node 101a can communicate with node 101b. But because there is no unidirectional path from node 101b to node 101a, this also implies that node 101b cannot communicate with node 101a. In contrast, bidirectional path 106 implies that node 101a can communicate with node 101d and vice-versa.

[033] Various mechanisms exist which allow one node to discover the network address of another node. A network administrator may provide node 101a with the network address of other reachable network nodes, for example through a configuration file, a command line interface, or a network management system such as SNMP. However, this mechanism is not feasible for today's networks which contain dozens or even hundreds of nodes.

[034] A node may also advertise its own network address to other reachable nodes by transmitting on routed network 102 using a broadcast or multicast address. A broadcast address is a single well-known address which all network nodes are aware of and can listen on. A multicast address is one of a group of well-known addresses which all network nodes are aware of and can listen on. Therefore, a node listening on either the broadcast address or one of the multicast addresses can learn the address of the sender without knowing the sender's address ahead of time. However, for a large scale network with hundreds of nodes broadcasting, this mechanism places a heavy burden on the nodes which receive large numbers of broadcast packets.

[035] FIG. 2 is a block diagram of one embodiment of a node 101 from FIG. 1 which utilizes the systems and/or methods for automatic discovery of network addresses. Node 101 contains a number of components that are well known in the art of data communications, including a processor 201, network interface 202, and memory 203. These components are coupled via bus 204. Omitted from FIG. 2 for simplicity are a number of conventional components that are not necessary to explain the operation of the system and/or method for automatic discovery of network addresses and known to those skilled in the art.

Network interface 202 provides communication with routed network 102 through link 104. Contained within memory 203 is address discovery logic 205. Address discovery logic 205 is configured to enable and drive processor 201 to allow the discovery of network addresses of nodes 101 on routed network 102. Memory 203 also contains a list of known nodes 206 containing the network address 103 of each node 101 in network 100 which this particular node 101 knows about.

[036] In some embodiments, each entry in the list 206 of known nodes contains a type field 207, a network address field 208, and a forward count field 209. In this example, there are only four entries, but any number of entries could be supported. Type field 207 is "discovered" if the entry was added through automatic discovery (discussed later). Type field 207 is "static" if the entry was added through add/delete mechanism 210, which can take the form of, for example, a configuration file, a command line interface or a network management system.

[037] The forward count field 209 controls how the network address field 208 of the node entry is forwarded to other nodes. If the forward count field 209 is zero, then the address is not forwarded to other nodes. If the forward count field 209 is non-zero, then the address is forwarded to other nodes for the number of hops in the forward count field 209. Use of the forward count field 209 is explained in detail later.

[038] The list of known nodes 206 is accessed by announcer logic 211, listener logic 212 and forwarder logic 213. Announcer logic 211 makes announcements to other nodes 101a-d about the network address 103 of those nodes it knows about, that is, the nodes in the list of known nodes 206. More specifically, announcer logic 211 makes announcements to listener logic 212 in other nodes 101. That is, listener logic 212 is listening for announcements from other announcers 211, where the announcements contain network addresses 103. On receipt of an announcement, listener logic 212 adds the network address 103 contained in the announcement to its own list of known nodes 206. Forwarder logic 213 may forward the announcement (received from another node 101) to all the nodes in its own list of known nodes 206, depending on the contents of the forward count field 209.

[039] In other embodiments, two lists are used rather than one. One list contains only Discovered nodes added through automatic discovery. The other list contains only Static

nodes added through add/delete mechanism 210. While the following descriptions will refer only to a single-list embodiment, one skilled in the art will recognize how the single list of known nodes 206 containing both Static and Discovered nodes of the first described embodiment could be adapted to work with another embodiment using two separate Static and Discovered lists.

[040] The workings of the systems and methods for automatic discovery of network addresses are explained by FIGs. 3A-C, which are a sequence of diagrams showing how the list of known nodes 206 in each node 101a-d is updated by announcements and forwarding. In explaining FIGs. 3A-C, reference will also be made to FIGs. 4 and 5, which are flowcharts of an embodiment of a method for automatic discovery of network addresses.

[041] In FIG. 3A, a network with configuration 100a has four nodes 101a-d connected to routed network 102. For simplicity, links 104a-d connecting the nodes to routed network 102 are not shown. Each node 101 is reachable by all of the other nodes through routed network 102. In this example, nodes 101a-d have network addresses .1, .2, .3 and .4, respectively. This minimal network address format is used for illustration only; actual network layer protocols such as IP use a more complicated format.

[042] The process of automatic discovery begins with step 401 in the flowchart of FIG. 4, where a node 101 initializes its list of known nodes 206. (These initial values are provided by the network administrator, through add/delete mechanism 210.) FIG. 3A shows a snapshot of each node's list of known nodes 206 after the initialization step 401 has been performed by each of the nodes 101a-d. The list of known nodes 206a for node 101a contains four entries, one for itself and one for each of the three remaining nodes. Each entry has type field 207 set to Static, meaning that the entries were not discovered by announcements received or forwarding by other nodes. Each entry has forward count field 209 set to nonzero, meaning that the entries will be forwarded to other nodes. The list of known nodes 206b-d for nodes 101b-d is empty.

[043] FIG. 3A also shows any paths which exist between nodes 101a-d at this initial time, as implied by the contents of the lists 206a-d. The entries in the list of known nodes 206a imply that a unidirectional path exists from node 101a to each of the other nodes

101b-d, as shown by dotted lines 301, 302, 303. There are no paths shown leading out of nodes 101b-d because the corresponding list for each of those nodes, 206b-d, is empty.

[044] The process of automatic discovery continues with step 402 in the flowchart of FIG. 4, where the current node in list 206 is examined. At step 403, if the type is Discovered then the next node is processed by continuing at step 407. If the type is Static, however, the current node is processed by continuing to step 404.

[045] At step 404, the forward count field 209 of the current node is compared to zero, and if the forward count field 209 is equal to zero, then the next node is processed by continuing at step 407. If the forward count field 209 is non-zero, the forward count field 209 is decremented at step 405. Next, at step 406, the network address field 208 and the (decremented) forward count field 209 of this current node is transmitted to all those nodes 101a-d in list 206 which have type Static. This transmission of a network address and a forward count from one node to another, based on a Static type node, is called an "announcement."

[046] Step 407 advances to the next node in the list 206, then step 408 determines if the newly advanced current node entry has reached the end of the list of known nodes 206. If the end of the list has not been reached, then processing continues back at step 402, where the newly advanced current node entry is examined. If the end of the list has been reached, step 409 informs each node to which an announcement was sent that announcements are finished. One skilled in the art should realize that the details of this step may be varied, depending on the implementation. For example, instead of waiting for all announcements to all nodes to be finished before sending an "announcement finished" to all nodes, it may be desirable to send an "announcement finished" to node X on completion of all announcements to node X, without waiting until all announcements to other nodes are complete.

[047] Processing of announcing continues at step 410, which waits for addition or deletion of a Static node by the network administrator. When a Static node is added or deleted, processing of the list begins again at step 402.

[048] FIG. 5 is a flow chart showing how announcements made by one node 101 are received and processed by another node 101. Step 501 waits for an announcement to be received. Step 502 determines whether the packet is an announcement or an

announcement-finished. If the packet type is announcement, processing continues at step 503, where a new entry for list of known nodes 206 is created. The new entry's network address field 208 and forward count field 209 are initialized at step 504 from the values in the announcement packet, and the new entry's type field 207 is set to Discovered. At step 505 the newly created node is added to list of known nodes 206.

[049] One skilled in the art should realize that no new information is imparted when an announcement is received by a node and the announcement contains the node's own network address. Therefore, the announcement can be ignored, or the announcer could avoid sending this packet by comparing the destination network address of the packet to the network address field 208.

[050] If step 502 determined that the received packet was an announcement-finished, then processing continues at step 506, where new node(s) are transmitted to all nodes 101a-d in list 206. Processing then continues again at step 501 where another announcement is waited on.

[051] This transmission of a network address and a forward count from one node to another, based on a Discovered node, is called a "forward." Forwarding differs from announcing in several ways. Announcing notifies other nodes about Static nodes but not about Discovered nodes, while forwarding notifies other nodes about both Static and Discovered nodes. Forwarding happens in response to receiving an announcement packet. Announcing happens at initialization, and in response to a new Static node being added/deleted by the add/delete mechanism 210.

[052] FIG. 3B follows FIG 3A in time. FIG. 3B shows the result of the first iteration of steps 402-407 and the first iteration of steps 501-508 for the example network. The first network address (.1) in the list 206a is announced to all nodes in the list (.1, .2, .3, .4), along with a decremented Forward Count of zero. To make this announcement to the first node in the list 206a, announcer logic 211 in node 101a constructs a packet containing, as data, the first network address (.1). The announcer logic 211 sets the destination network address of this packet to .1 and transmits the packet. To make this announcement to the second node in the list 206a, announcer logic 211 in node 101a transmits a packet with the same data and with a destination network address of .2.

Similarly, announcements are made to the third and fourth nodes in the list using packets with destination network addresses of .3 and .4.

[053] Making announcements to specific nodes using unicast destination network addresses is preferable to making an announcement to all nodes using the broadcast destination network address, because all nodes on the network, even those not interested in participating in Discovered discovery, receive broadcast packets. The broadcasting technique is used by other protocols (such as routing protocols), so that network nodes already have to process broadcast traffic. The system and method of the present invention does not require each node to process increased broadcast traffic. Using unicast destination network addresses is also preferable to using a multicast address. Multicast packets are not received by all nodes, only those listening to the particular multicast address, but multicast is not widely supported by network layer protocols. The use of unicast addresses is especially desirable for carrying network management traffic which is in-band.

[054] As shown in FIG. 3B, the lists 206b-d for the other nodes are updated with the announced address of .1, so that the lists 206b-d which were empty in FIG. 3A now contain one entry each. This single entry has a network address field 208 of .1, a forward count field 209 of zero (decremented from its original value in list 206a), and a type field 207 of Discovered, since the network address was discovered by announcement. The forwarders 213 in nodes 101b-d do not forward the announcement packets after receipt by listener logic 212 in the same node, since the forward count field 209 of the received packets are zero.

[055] As a result of the updated lists 206b-d, by which nodes 101b-d learned the network address of node 101a, a path now exists from each of nodes 101b-d to 101a. FIG. 3B thus shows paths 301, 302, 303 as bidirectional, where in FIG. 3A these same paths were unidirectional.

[056] The above description, using one network address per announcement, results in 16 announcement packets for the example network configuration. One skilled in the art should recognize that an announcement packet destined for a particular node can contain as data the network address of more than one node. In other embodiments, a single announcement packet containing the network address and forward count of each node in

the list of known nodes 206 could be transmitted to each of the four nodes in the list. In the example network configuration, this results in four announcement packets rather than 16. One skilled in the art should recognize that many packet sizes and thus many numbers of network addresses in each packet are possible.

[057] FIG. 3C follows FIG. 3B in time, and shows the result of all iterations of steps 402-407 and all iterations of steps 501-508 for the example network configuration. In FIG. 3C, node 101a has announced each network address in its list of known nodes 206a (.1, .2, .3 and .4) to all nodes 101a-d (.1, .2, .3 and .4). The nodes have received the announcements and added corresponding entries in their own lists 206b-d, such that nodes 101b-d now know about all nodes 101a-d. The forwarders 213 in nodes 101b-d did not forward the announcement packets after receipt by listener logic 212 in the same node, since the forward count field 209 of the received packets are zero.

[058] Lists 206b-d differ from list 206a. The entries in list 206a are of type Static, while the entries in lists 206b-d are of type Discovered. Also, the entries in lists 206b-d have Forward Counts of zero, since the original Forward Count of 1 in list 206a was decremented before transmitting to the other nodes in the announcement. However, the Forward Count in the sender's list 206 is not decremented.

[059] As a result of the updated lists 206b-d, by which nodes 101b-d learned the network addresses of node 101a-d, a path now exists from each of nodes 101a-d to each of the other nodes 101a-d. This arrangement is called a "full mesh." FIG. 3C thus shows the same three bidirectional paths 301, 302, 303 from FIG. 3B, plus two additional bidirectional paths 304 and 305.

[060] The workings of one embodiment of a system and method for automatic discovery of network addresses are further explained by FIGs. 6A-D, which are a sequence of diagrams showing the announcements and forwarding for a different network configuration 100b. As with FIGs. 3A-C, FIGs. 6A-D show how the list of known nodes 206 in each node 101a-d is updated by announcements and forwarding.

[061] The network configuration 100b (FIGs. 6A-D) is different than network configuration 100a (FIGs. 3A-C). Network configuration 100b has seven nodes rather than four, and uses three routed networks 102a-c instead of one. The list of known nodes 206 used with network configuration 100b is also different. Network configuration 100a

contained a one list, 206a, which contained the node addresses of all other nodes. In contrast, the node addresses in network configuration 100b are distributed into three levels. List 206a is the first level. List 206a has node addresses for two nodes, forming a second level. List 206b and list 206c have node addresses for the remaining four nodes, forming a third level.

[062] FIG. 6A shows a snapshot of each node's list of known nodes 206 after the initialization step 401 has been performed by each of the nodes 101a-g. The list of known nodes 206a for node 101a contains two entries, one for node 101b and one for node 101b. Both entries have the forward count field 209 set to 1 and the type field 207 set to Static. This set of entries for list 206a implies the existence of unidirectional paths 601 and 602.

[063] The list of known nodes 206b for node 101b contains two entries, one for node 101d and one for node 101e. Both entries have the forward count field 209 set to 1 and the type field 207 set to Static. This set of entries for list 206b implies the existence of unidirectional paths 603 and 604.

[064] The list of known nodes 206c for node 101c contains two entries, one for node 101f and one for node 101g. Both entries have the forward count field 209 set to 1 and the type field 207 set to Static. This set of entries for list 206c implies the existence of unidirectional paths 605 and 606.

[065] FIG. 6B shows a snapshot of each node's list of known nodes 206 after node 101a has announced the Static nodes in its list 206a. There are two Static nodes in list 206a, and both have a non-zero Forwarding Count. Therefore, node 101a announces address .2 to both Static nodes in its list 206a (.2 and .3), and also announces address .3 to both Static nodes.

[066] The list 206b in node 101b remains unchanged by the announcement of node .2, since the announcement contained only the node's own address. However, the list 206c in node 101c has been updated with a new Discovered node with the network address field 208 set to .2, and with forward count field 209 set to zero. Because forward count field 209 is zero (node 101a decremented the count before transmitting the announcement), node 101c does not forward the new node address (.2) on to other nodes.

[067] The list 206c in node 101c remains unchanged by the announcement of node .3, since the announcement contained only the node's own address. However, the list 206b in node 101b has been updated with a new Discovered node with the network address field 208 set to .3, and with forward count field 209 set to zero. Because forward count field 209 is zero (node 101a decremented the count before transmitting the announcement), node 101b does not forward the new node address (.3) on to other nodes. The two new Discovered nodes in list 206b and list 206c implies the existence of two unidirectional paths, which is equivalent to bidirectional path 607 as shown.

[068] FIG. 6C shows a snapshot of each node's list of known nodes 206 after node 101b has announced the Static nodes in its list 206b. There are two Static nodes in list 206b, and both have a non-zero Forwarding Count. Therefore, node 101b announces address .4 to both Static nodes in its list 206b (.4 and .5), and also announces address .5 to both Static nodes.

[069] The list 206d in node 101d remains unchanged by the announcement of node .4, since the announcement contained only the node's own address. However, the list 206e in node 101e has been updated with a new Discovered node with the network address field 208 set to .4, and with forward count field 209 set to zero. Because forward count field 209 is zero (node 101a decremented the count before transmitting the announcement), node 101e does not forward the new node address (.5) on to other nodes.

[070] The list 206e in node 101e remains unchanged by the announcement of node .5, since the announcement contained only the node's own address. However, the list 206d in node 101d has been updated with a new Discovered node with the network address field 208 set to .5, and with forward count field 209 set to zero. Because forward count field 209 is zero (node 101a decremented the count before transmitting the announcement), node 101d does not forward the new node address (.5) on to other nodes. The two new Discovered nodes in list 206d and list 206e implies the existence of two unidirectional paths, which is equivalent to bidirectional path 608 as shown.

[071] FIG. 6D shows a snapshot of each node's list of known nodes 206 after node 101c has announced the Static nodes in its list 206b. There are two Static nodes in list 206c, and both have a non-zero Forwarding Count. Therefore, node 101c announces address .6

to both Static nodes in its list 206c (.6 and .7), and also announces address .6 to both Static nodes.

[072] The list 206f in node 101d remains unchanged by the announcement of node .6, since the announcement contained only the node's own address. However, the list 206g in node 101g has been updated with a new Discovered node with the network address field 208 set to .6, and with forward count field 209 set to zero. Because forward count field 209 is zero (node 101a decremented the count before transmitting the announcement), node 101e does not forward the new node address (.6) on to other nodes.

[073] The list 206g in node 101g remains unchanged by the announcement of node .7, since the announcement contained only the node's own address. However, the list 206f in node 101f has been updated with a new Discovered node with the network address field 208 set to .7, and with forward count field 209 set to zero. Because forward count field 209 is zero (node 101a decremented the count before transmitting the announcement), node 101d does not forward the new node address (.7) on to other nodes. The two new Discovered nodes in list 206f and list 206g implies the existence of two unidirectional paths, which is equivalent to bidirectional path 609 as shown.

[074] The paths resulting from the sequence described by FIGs. 6A-D a is called "dual tier." FIG. 6D thus shows the same three bidirectional paths 301, 302, 303 from FIG. 3C, plus two additional bidirectional paths 306 and 307.

[075] The workings of an embodiment of a system and method for automatic discovery of network addresses are further explained by FIGs. 7A-D, which are a sequence of diagrams showing the announcements and forwarding for a different network configuration 100c. As before, FIGs. 7A-D show how the list of known nodes 206 in each node 101a-g is updated by announcements and forwarding.

[076] The network configuration 100c (FIGs. 7A-D) is similar to network configuration 100b (6A-D), with seven nodes in three levels. But there are noticeable differences in the lists 206: the lists 206 in network configuration 100c contain some Static nodes with Forward Counts set to 2. This difference results in a different set of paths being created during the automatic discovery process, as will be shown in FIG. 7D.

[077] FIG. 7A shows a snapshot of each node's list of known nodes 206 after the initialization step 401 has been performed by each of the nodes 101a-g. The list of

known nodes 206a for node 101a contains two entries, one for node 101b and one for node 101b. Both entries have the type field 207 set to Static and the forward count field 209 set to 2. This set of entries for list 206a implies the existence of unidirectional paths 601 and 602.

[078] The list of known nodes 206b for node 101b contains two entries, one for node 101d and one for node 101e. Both entries have the type field 207 set to Static. One entry has the forward count field 209 set to 2 and the other has the forward count field 209 set to 1. This set of entries for list 206b implies the existence of unidirectional paths 603 and 604.

[079] The list of known nodes 206c for node 101c contains two entries, one for node 101f and one for node 101g. Both entries have the type field 207 set to Static. One entry has the forward count field 209 set to 2 and the other has the forward count field 209 set to 1. This set of entries for list 206c implies the existence of unidirectional paths 605 and 606.

[080] FIG. 7B shows a snapshot of each node's list of known nodes 206 after node 101a has announced the Static nodes in its list 206a. There are two Static nodes in list 206a, and both have a non-zero Forwarding Count. Therefore, node 101a announces address .2 to both Static nodes in its list 206a (.2 and .3), and also announces address .3 to both Static nodes.

[081] The list 206b in node 101b remains unchanged by the announcement of node .2, since the announcement contained only the node's own address. However, the list 206c in node 101c has been updated with a new Discovered node with the network address field 208 set to .2, and with forward count field 209 set to 1. (Forwarding of this new non-zero Forward Count node will be described later, with reference to FIG. 7D.)

[082] The list 206c in node 101c remains unchanged by the announcement of node .3, since the announcement contained only the node's own address. However, the list 206b in node 101b has been updated with a new Discovered node with the network address field 208 set to .3, and with forward count field 209 set to 1. The two new Discovered nodes in list 206b and list 206c implies the existence of two unidirectional paths, which is equivalent to bidirectional path 607 as shown.

[083] FIG. 7C shows a snapshot of each node's list of known nodes 206 after node 101b and node 101c have both announced the Static nodes in their lists 206b. There are two Static nodes in each of the lists 206b and 206c, and both have a non-zero Forwarding Count. Therefore, node 101b announces address .4 to both Static nodes in its list 206b (.4 and .5), and also announces address .5 to both Static nodes. Similarly, node 101c announces address .6 to both Static nodes in its list 206c (.6 and .7), and also announces address .7 to both Static nodes.

[084] The result of the announcement of addresses .4 and .5 by node 101b is as follows. The list 206e in node 101e is updated with a new Discovered node with the network address field 208 set to .4, and with forward count field 209 set to zero. The list 206d in node 101d is updated with a new Discovered node with the network address field 208 set to .5, and with forward count field 209 set to zero. Because the forward count fields on these new nodes are zero (decremented from 1 before transmitting the announcement), node 101d and node 101e do not forward the new node addresses (.4 and .5) on to other nodes. The two new Discovered nodes in list 206d and list 206e implies the existence of two unidirectional paths, which is equivalent to bidirectional path 608 as shown.

[085] Similarly, the result of the announcement of addresses .6 and .7 by node 101c is as follows. The list 206g in node 101g is updated with a new Discovered node with the network address field 208 set to .6, and with forward count field 209 set to zero. The list 206f in node 101f is updated with a new Discovered node with the network address field 208 set to .7, and with forward count field 209 set to zero. Because the forward count fields on these new nodes are zero (decremented from 1 before transmitting the announcement), node 101f and node 101g do not forward the new node addresses (.6 and .7) on to other nodes. The two new Discovered nodes in list 206f and list 206g implies the existence of two unidirectional paths, which is equivalent to bidirectional path 609 as shown.

[086] FIG. 7D shows a snapshot of each node's list of known nodes 206 after node 101b and node 101c have forwarded the Discovered nodes in their lists 206b and 206c. Node 101b announces address .3 to all other nodes in its list 206b, which is nodes .4 and .5. Node 101c announces address .2 to all other nodes in its list 206c, which is nodes .6 and .7.

[087] The result of the forwarding of address .3 by node 101b is as follows. The list 206d in node 101d is updated with a new Discovered node with the network address field 208 set to .3, and with forward count field 209 set to zero. The new Discovered node in list 206d implies the existence of a unidirectional path 710 from node 101d to node 101c. The list 206e in node 101e is updated with a new Discovered node with the network address field 208 set to .3, and with forward count field 209 set to zero. The new Discovered node in list 206e implies the existence of a unidirectional path 711 from node 101e to node 101c.

[088] The result of the forwarding of address .2 by node 101c is as follows. The list 206d in node 101f is updated with a new Discovered node with the network address field 208 set to .2, and with forward count field 209 set to zero. The new Discovered node in list 206f implies the existence of a unidirectional path 712 from node 101f to node 101b. The list 206g in node 101g is updated with a new Discovered node with the network address field 208 set to .2, and with forward count field 209 set to zero. The new Discovered node in list 206g implies the existence of a unidirectional path 713 from node 101g to node 101b.

[089] Some Discovered nodes in FIG. 7D have a forward count field 209 of .1. However, no further sequence diagrams are needed because none of the lists 206 will change after these Discovered nodes are forwarded. This is true for this particular network configuration 100c because the forwarded nodes will be duplicates of Discovered nodes already in the list.

[090] The set of paths created by network configuration 100c, as shown in FIG. 7D, is different than the set of paths created by network configuration 100b, as shown in FIG. 6D. In FIG. 7D, each node in the lowest level is connected to both nodes in the second level, by paths 603, 604, 710, 711 and 605, 606, 712, 713. In contrast, in FIG. 6D, each node in the lowest level is connected to only one node in the second level, by paths 603, 604 and 605, 606. There is no equivalent to paths 710, 711, 712 and 713 in FIG. 6D.

[091] FIG. 8 shows the structure of an announcement packet used in an embodiment of a system and method for automatic discovery of network nodes. The announcement packet 801 comprises a number of fields: special identifier 802; number of messages 803; length 804; repeat count 805; message type 806; and message data 807.

[092] The announcement packet 801 is encapsulated in an ICMP packet 808. That is, the announcement packet 801 is contained within the data field of the ICMP packet 808. An ICMP packet 808 comprises a number of fields: type 809; code 810; checksum 811 and data 812.

[093] The ICMP packet 808 is itself encapsulated in an IP packet 813. That is, the ICMP packet 808 is contained within the data field of the IP packet 813. IP packet 813 will not be completely described here. Instead, only those fields in IP packet 813 which the system and method of the present invention affect will be described. Those fields are: protocol 814; source network address 815; destination network address 816; and data 817.

[094] The announcement packet 801 is used as follows. One or more node addresses to be announced are put in the message data field 807. Repeat count 805 is set to the number of node addresses contained in this announcement. Special identifier 802 is set to a predetermined value such as "FEED" which distinguishes this from a standard Ping packet. Message Type 806 is set to a predetermined value which identifies the packet as an announcement. The length field 804 is set to the total length of the packet, which depends on the number of node addresses contained within.

[095] The header of the ICMP packet 808 is filled in as follows. The ICMP type field 809 is set to 8 and the ICMP code field is set to 810, which identifies the as ICMP packet 808 an Echo Request. The ICMP checksum 811 is set to the calculated checksum of the entire ICMP packet.

[096] The header of the IP packet 813 is filled in as follows. The protocol field 814 is set to 1, which identifies the IP packet as encapsulating an ICMP packet 808. The source network address 815 is set to the network address 103 which is making the announcement. The destination network address 816 is set to the network address 103 of a particular node which is the target of the announcement. As explained before, this network address is a unicast address, so that the announcement packet 801 is delivered (by the IP protocol layer) only to that particular node.

[097] The announcement-finished packet (not shown) is similar, but message type 806 is set to a predetermined value which identifies the packet as an announcement-finished,

and message data 807 is not used. The header of the ICMP packet 808, and the header of the IP packet 813, are filled in the same manner as for the announcement packet 801.

[098] One skilled in the art should recognize that this is only one example of packet structures which could be used for announcements and announcement-finished, and that many other packet structures are possible.

[099] FIG. 9A-B illustrates another embodiment of the system and method for automatic discovery of network addresses. This embodiment detects when a node 101 becomes unreachable, and responds as follows. FIG. 9A shows network configuration 100d before the detection of an unreachable node, and FIG. 9B shows the configuration after detection.

[0100] In FIG. 9A, node 101b detects that node 101a is unreachable. In response to detection of unreachable node 101a, node 101b looks for entries in its list of known nodes 206b with a discovery source field 210 which matches the unreachable node 101a. (In nodes of Discovered type, the discovery source field 210 is set to the network address 103 of the node which sent the announcement.) Here the matching nodes have addresses .3 and .4.

[0101] In FIG. 9B, node 101b has deleted from its list 206 the entries corresponding to the matching node 101 (with addresses .3 and .4). Node 101b has also announced, to all other nodes, the deletion of the matching nodes (with addresses .3 and .4). On receipt of a deletion announcement, a node 101 deletes from its list of known nodes 206 the node addresses in the announcement. Here, node 101c has deleted address .4, but the deletion announcement for its own address .3 has no effect. Node 101 has deleted address .3 but the deletion announcement for its own address .4 has no effect. After receiving a deletion announcement, a node 101 also forwards the announcement to all other nodes in its list 206, if the forward count field 209 is non-zero.

[0102] In one embodiment, an unreachable node 101 is detected by receiving a deletion event from add/delete mechanism 210. In another embodiment, an unreachable node 101 is detected through a polling mechanism which periodically sends a poll packet to each node 101 in list 206 and receives a response in return. If a particular node 101 stops responding to polls, then the node is considered unreachable by the poll sender. One skilled in the art will recognize that the polling mechanism can be implemented in

various ways. For example, a responder can be considered unreachable after a single response is missed, or after more than one response is missed. In some embodiments an aging technique is used, such that when a first poll response is missed, the poller records the event and/or the time the response was missed. If the poller sends out subsequent polls that are also missed, the poller ages the matching nodes again to update the event and/or time the response is missed. When a predetermined number of polls are missed, the poller determines that the node with missing responses is unreachable, and proceeds as described above.

[0103] The foregoing description has been presented for purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise forms disclosed. Obvious modifications or variations are possible in light of the above teachings. The embodiments discussed, however, were chosen and described to illustrate the principles of the invention and its practical application to thereby enable one of ordinary skill in the art to utilize the invention in various embodiments and with various modifications as are suited to the particular use contemplated. All such modifications and variation are within the scope of the invention as determined by the appended claims when interpreted in accordance with the breadth to which they are fairly and legally entitled.